CSE574 Introduction to Machine Learning

Advance Practices and Unsupervised Learning: k-nearest neighbors

Jue Guo

University at Buffalo

March 13, 2024

Outline

k-nearest neighbors

Learning Objective

- Define k-nearest neighbors.
- List the steps of the k-nearest neighbors algorithm.
- Explain how to choose an appropriate value of k.
- Use a decision boundary plot to explore different values of k.
- Define Euclidean distance, Minkowski distance, and Manhattan distance.

We will do a signup sheet to track attendance at the end of class today.

k-nearest neighbor

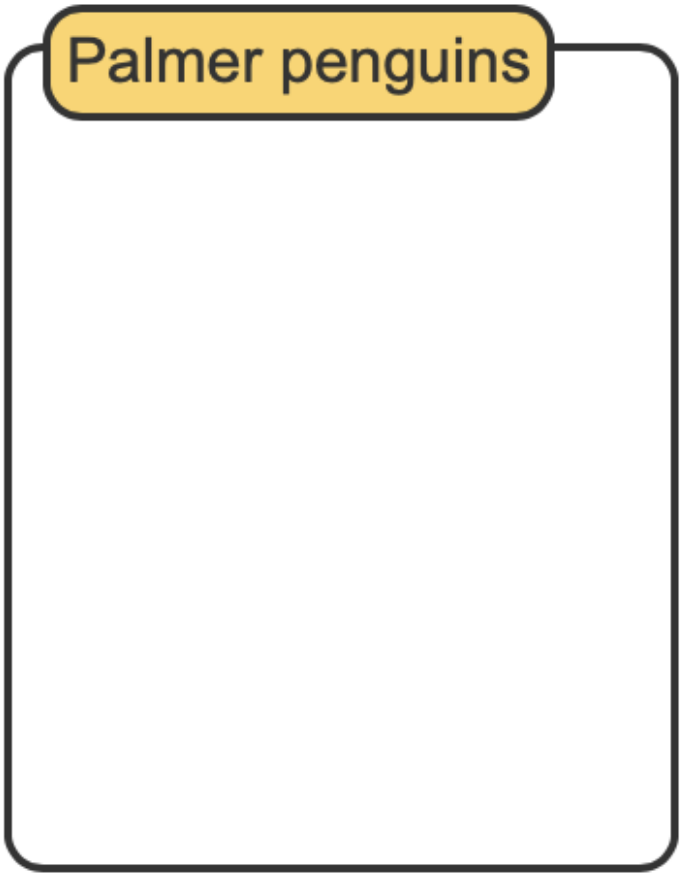
***k*-nearest neighbors** is a supervised classification algorithm that predicts the class of an output feature based on the class of other instances with the most similar, or "nearest," input features.

- The *k* nearest instances, or neighbors, are identified using some distance measure, and the classes of each neighbor's output feature are identified.

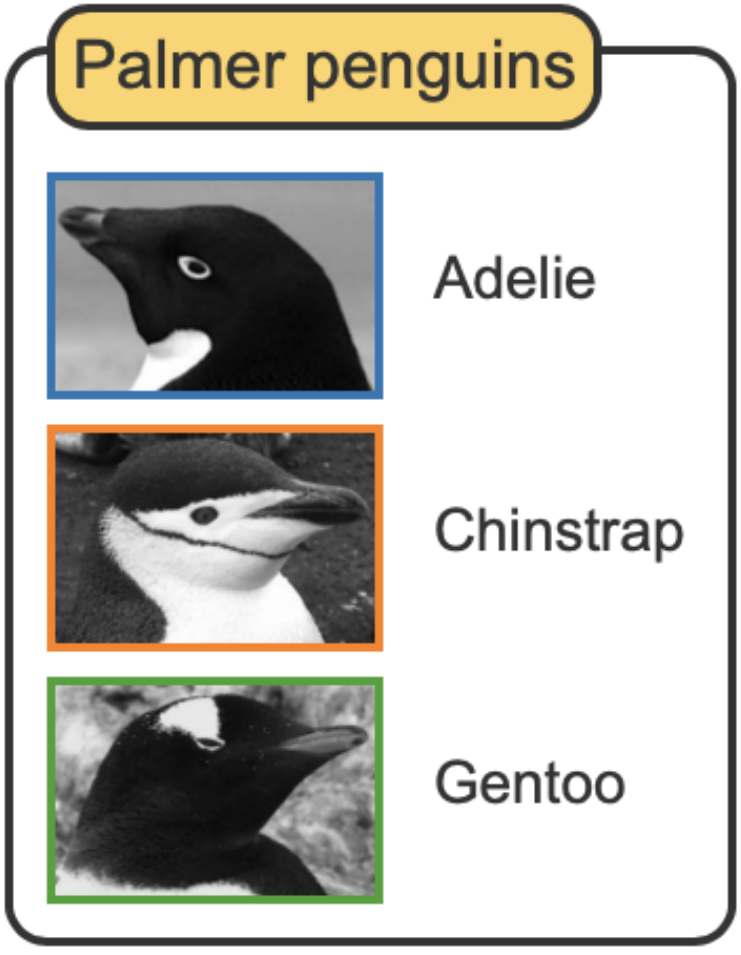
The most frequently occurring class from the *k* closest instances becomes the prediction.

Let's look at an example for k-nearest neighbor;

An Example: Classifying penguins based on bill length:

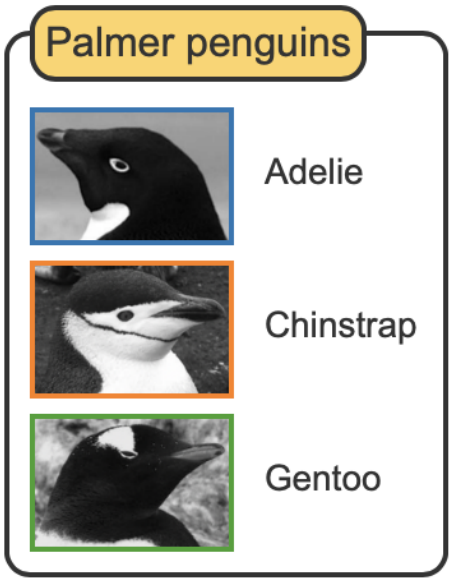


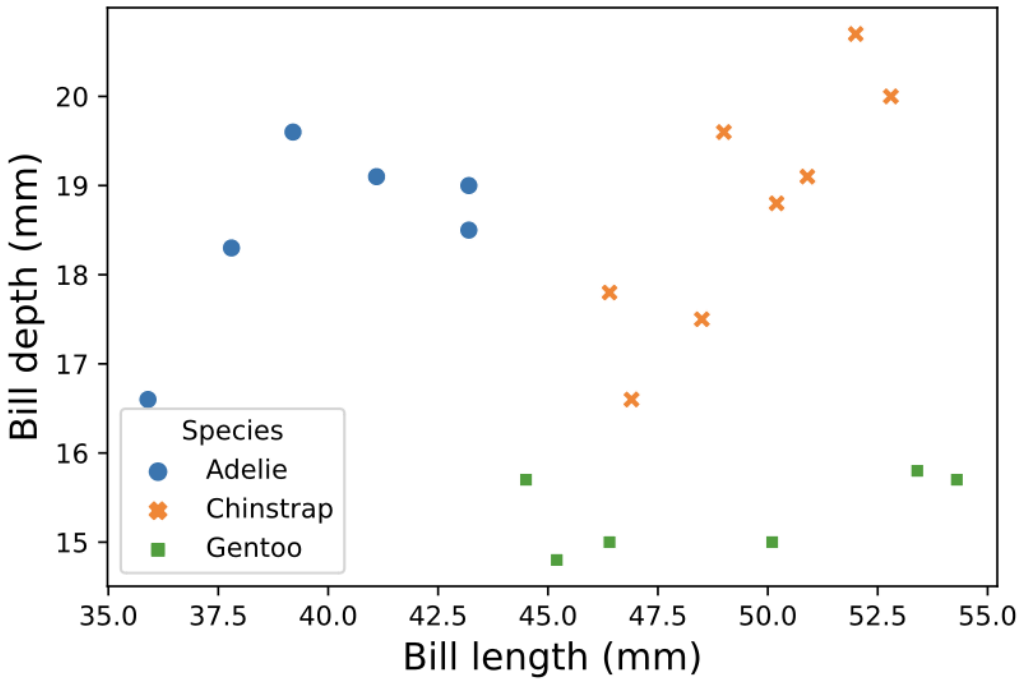
- The Palmer penguins dataset was collected by researchers studying penguins on the Palmer Archipelago in Antarctica.



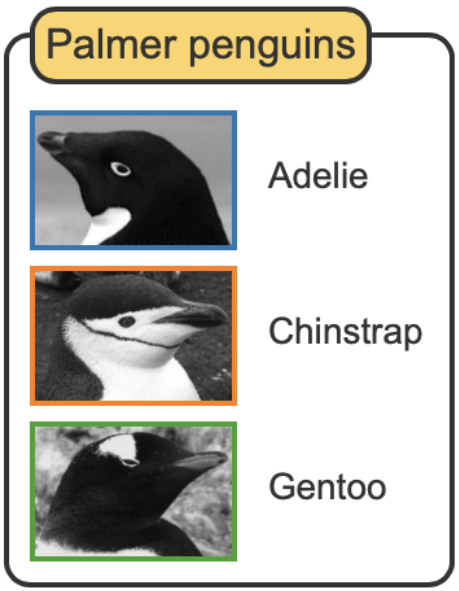
- Three species of penguins were studied: Adelie, Chinstrap, and Gentoo.

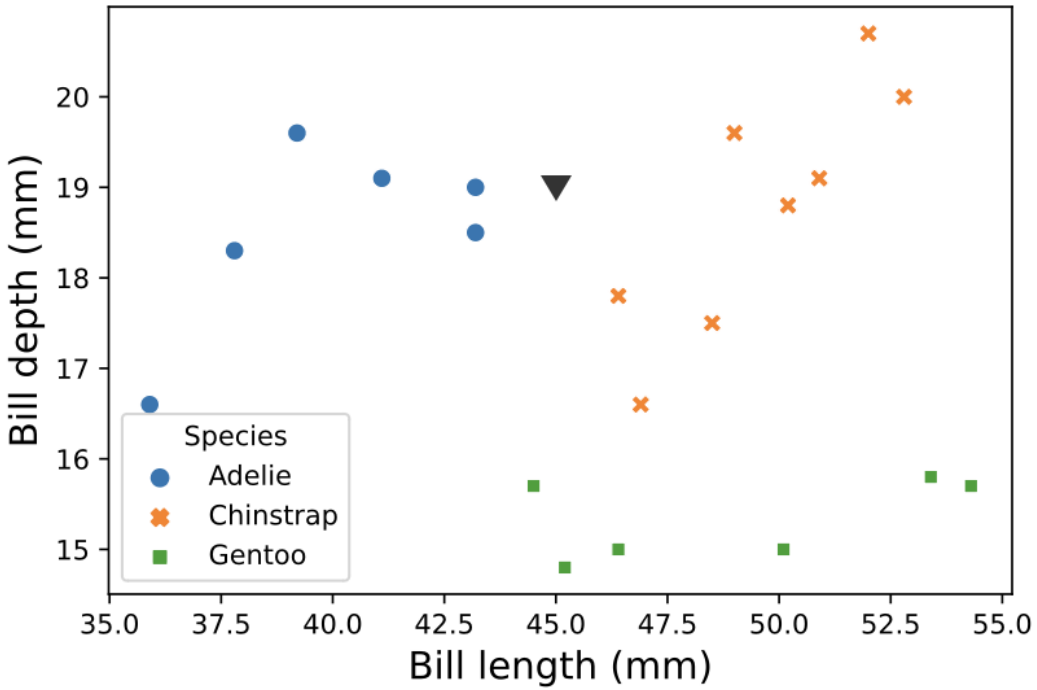
- Body measurements were taken from each penguin, including bill length and bill depth.



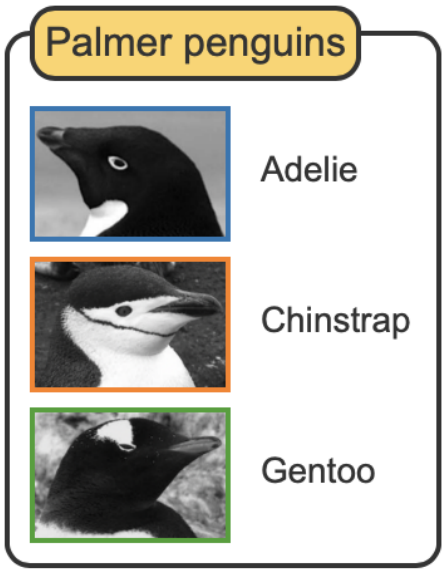


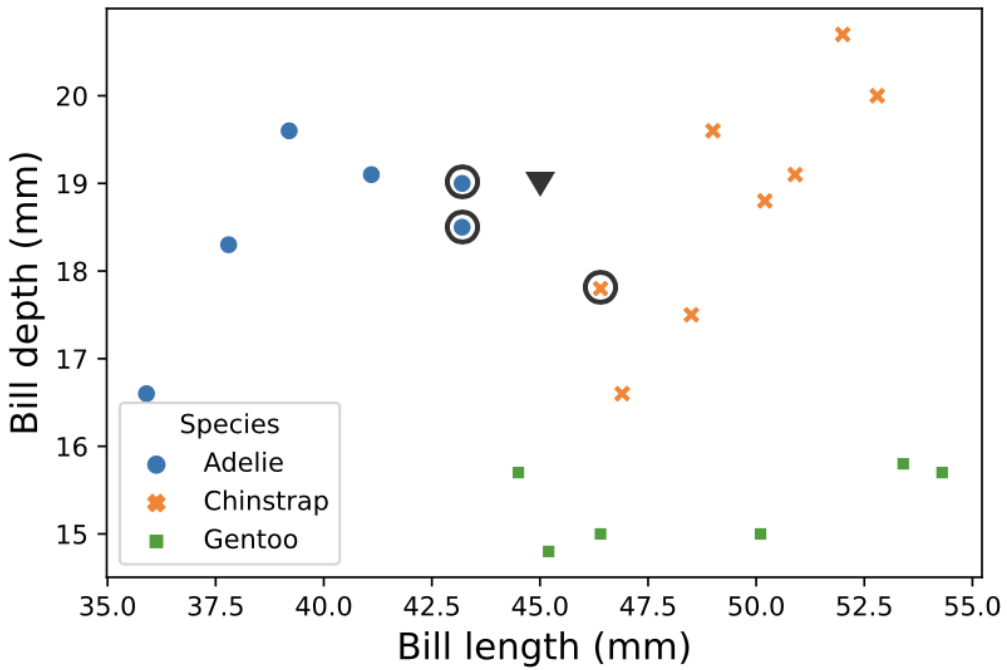
- Suppose an unknown penguin has a bill length of 45 mm and a bill depth of 19 mm. Which species is most likely for this penguin?



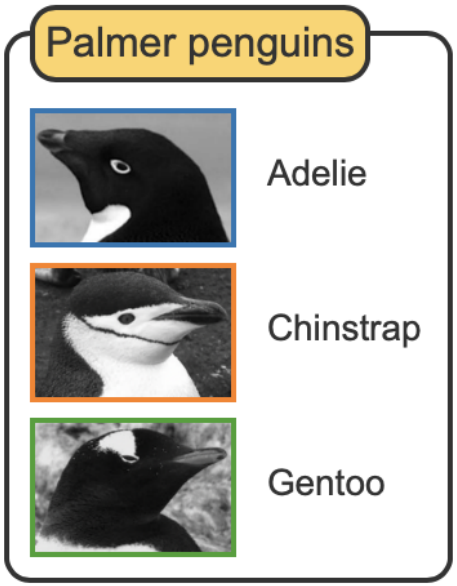


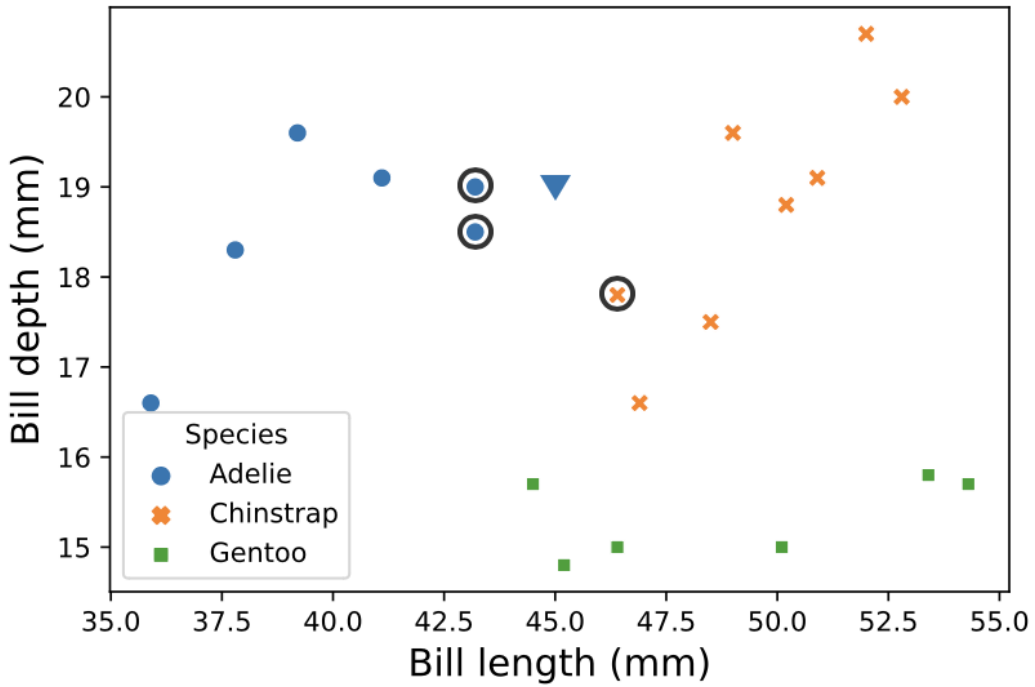
- The three nearest penguins to (45, 19) are identified. Two are Adelie penguins, and one is a Chinstrap penguin.





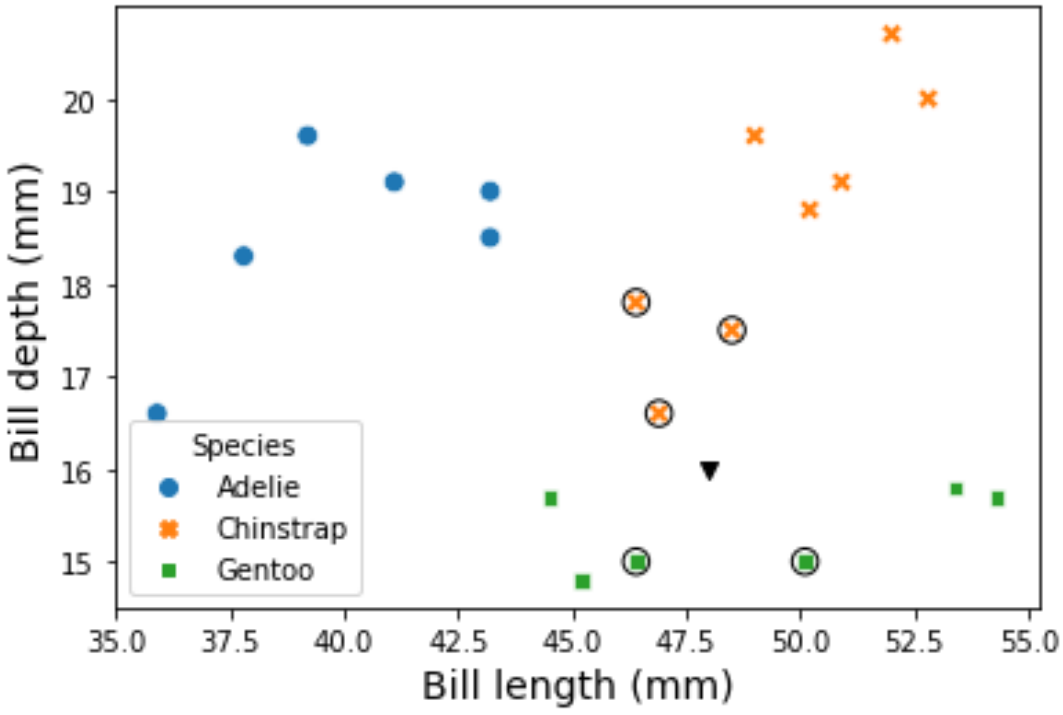
- 2/3 = 67 % of the nearest penguins are Adelie. So, a penguin at (45, 19) is predicted to be an Adelie penguin.





Practice Questions

A penguin with unknown species has a bill length of 48 mm and a bill depth of 16 mm (black triangle). The five nearest neighbors for this penguin are circled in the scatter plot below.



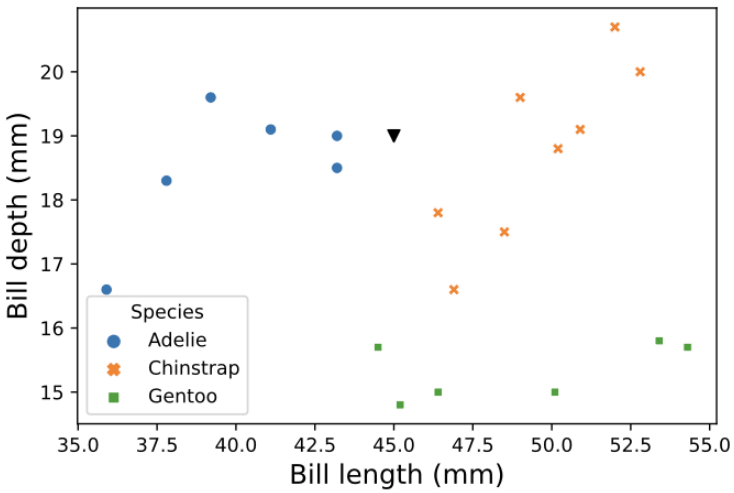
k-nearest neighbors algorithm

Let $\mathbf{x}_i = \left(x_{1i}, x_{2i}, \ldots, x_{pi}\right)$ denote the p input features from instance i, and let y_i denote the output feature. Let n denote the number of instances in a dataset. A new instance x^* is classified in k-nearest neighbors by identifying the k-nearest instances to x^*, and assigning the most common class as the prediction.

To classify a new instance, $\boldsymbol{x}^*$:

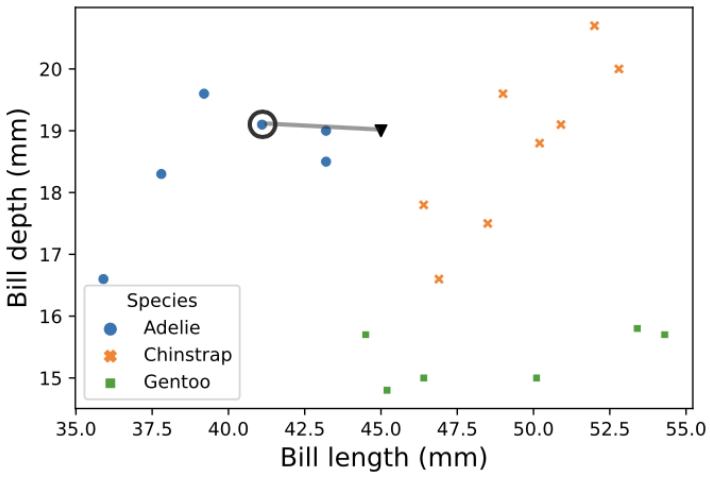
1. Select an integer value for $\boldsymbol{k}$.
2. Calculate the distance between $\boldsymbol{x}^*$ and all other $\boldsymbol{x_i}$ from $\boldsymbol{i = 1, \ldots, n}$.
3. Sort the distances from smallest to largest, and identify $\boldsymbol{i}$ for the $\boldsymbol{k}$ smallest distances. The instances with the smallest distances to $\boldsymbol{x}^*$ are the nearest neighbors.
4. Calculate the proportion of times each class occurs in the $\boldsymbol{k}$ nearest neighbors. For each class, the proportion of times the class occurs are the predicted probabilities of class membership.
5. The predicted class for instance $\boldsymbol{x}^*$ is the class that occurs most often in the nearest neighbors.

Applying the Algorithm

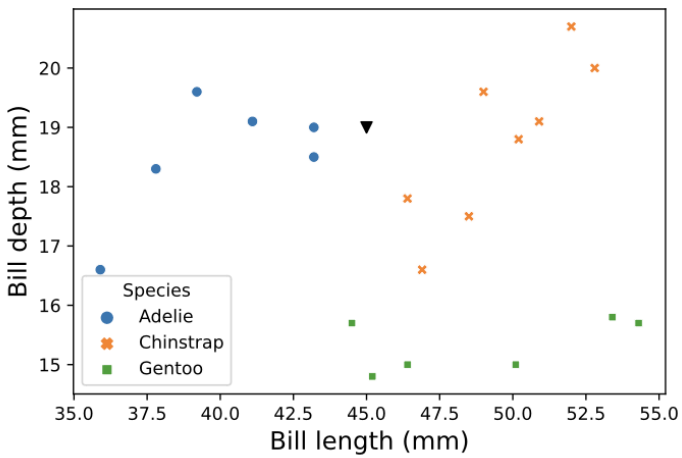


Consider a random sample of 20 penguins. Using k-nearest neighbors with k=3, how should a penguin at $x^* = (45, 19)$ be classified?

Instance	Distance
1	3.9

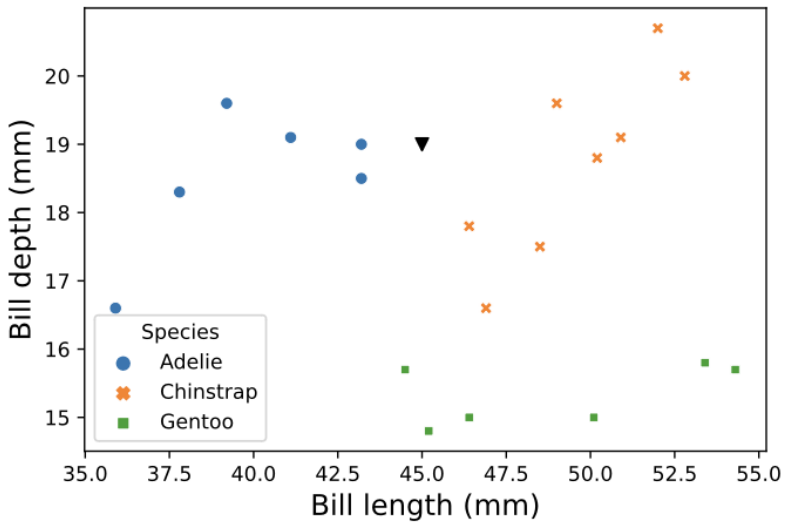


Instance 1 has a bill length of 41.1 mm and a bill depth of 19.1 mm. The distance between Instance 1 and x^* is 3.9.



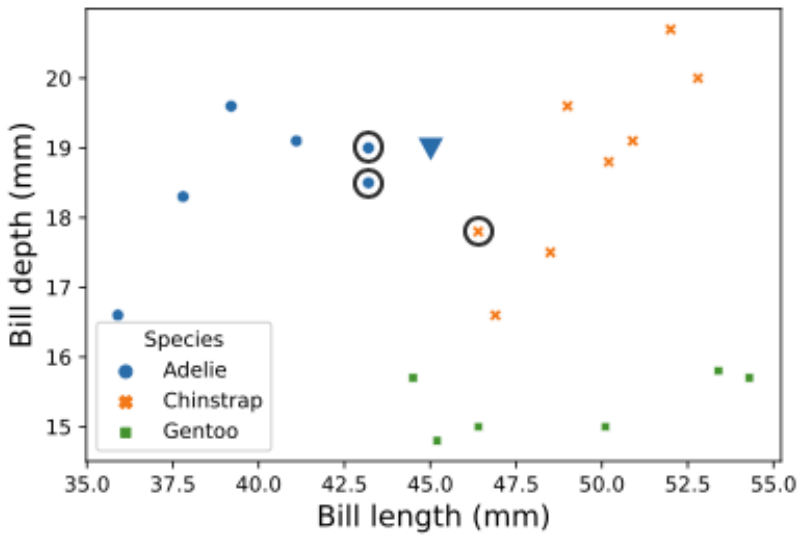
Instance	Distance
1	3.9
2	7.2
3	1.9
4	9.4
5	5.8
6	1.8
7	5.2
8	3.1
9	5.9
10	3.8
11	7.9
12	1.9
13	7.2
14	4.0
15	9.9
16	3.3
17	4.2
18	9.0
19	6.5
20	4.2

The distance between each instance and x^* is calculated and added to the list of distances.



Instance	Distance
6	1.8
12	1.9
3	1.9
8	3.1
16	3.3
10	3.8
1	3.9
14	4.0
20	4.2
17	4.2
7	5.2
5	5.8
9	5.9
19	6.5
13	7.2
2	7.2
11	7.9
18	9.0
4	9.4
15	9.9

The distances are sorted from smallest to largest, and the k=3 nearest neighbors are identified.



Instance	Distance	Class
6	1.8	Adelie
12	1.9	Adelie
3	1.9	Chinstrap
8	3.1	
16	3.3	
10	3.8	
1	3.9	
14	4.0	
20	4.2	
17	4.2	
7	5.2	
5	5.8	
9	5.9	
19	6.5	
13	7.2	
2	7.2	
11	7.9	
18	9.0	
4	9.4	
15	9.9	

The probability that a nearest neighbor is Adelie is 2/3 = 0.67. The probability that a nearest neighbor is Chinstrap is 1/3 = 0.33. Since Adelie has a greater probability, y^* is predicted Adelie.

Practice Round: Steps of the k-nearest neighbors algorithm.

Let x^* denote a new instance. Identify the steps for classifying x^* using k-nearest neighbors.

1) Step 1: _____

- ○ Calculate the distance between x^* and all other instances.
- ○ Select an integer value for k.
- ○ Select a value for k.

Let x^* denote a new instance. Identify the steps for classifying x^* using k-nearest neighbors.

1) Step 1: _____

- Calculate the distance between x^* and all other instances.
- Select an integer value for k.
- Select a value for k.

Correct

k must be integer-valued. Ex: x^* cannot have 2.5 nearest neighbors, but x^* can have 3 nearest neighbors.

2) Step 2: _____

- ○ Calculate the distance between x^* and all other instances.
- ○ Calculate the distance between all pairs of instances, not including x^*.
- ○ Calculate the distance between all pairs of instances, including x^*.

2) Step 2: _____

- (●) Calculate the distance between x^* and all other instances.
- ( ) Calculate the distance between all pairs of instances, not including x^*.
- ( ) Calculate the distance between all pairs of instances, including x^*.

Correct

Since x^* is the instance to classify, only the n distances between x^* and $x_1, \ldots, x_n$ are necessary.

3) Step 3: _____

- ○ Sort the output features from smallest to largest.
- ○ Sort the input features from smallest to largest.
- ○ Sort the distances from smallest to largest.

3) Step 3: _____

- ○ Sort the output features from smallest to largest.
- ○ Sort the input features from smallest to largest.
- ◉ Sort the distances from smallest to largest.

Correct

k-nearest neighbors classifies based on the k nearest instances. The k smallest distances belong to the nearest neighbors.

4) Step 4: _____

- ○ Calculate the proportion of times each class occurs in the k nearest neighbors.
- ○ Calculate the proportion of times each class occurs in the k farthest neighbors.
- ○ Calculate the proportion of times each class occurs in the full dataset.

5) Step 5: _____

- ○ Classify x^* as the majority class.
- ○ Classify x^* as the most frequently occurring class.

4) Step 4: ____

- (●) Calculate the proportion of times each class occurs in the k nearest neighbors.
- ( ) Calculate the proportion of times each class occurs in the k farthest neighbors.
- ( ) Calculate the proportion of times each class occurs in the full dataset.

Correct

The k nearest neighbors are the most similar instances to x^*. Only the nearest neighbors are used in the prediction.

5) Step 5: ____

- ( ) Classify x^* as the majority class.
- (●) Classify x^* as the most frequently occurring class.

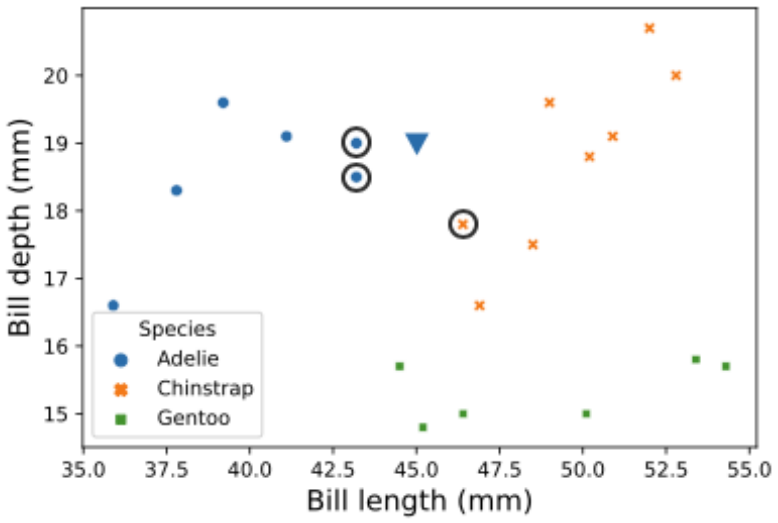
Correct

When the number of classes c is three or more, a majority may not exist. Ex: If 40% of the nearest neighbors are Adelie, 30% are Chinstrap, and 30% are Gentoo, the k-nearest neighbors model will predict Adelie.

Selecting an appropriate k

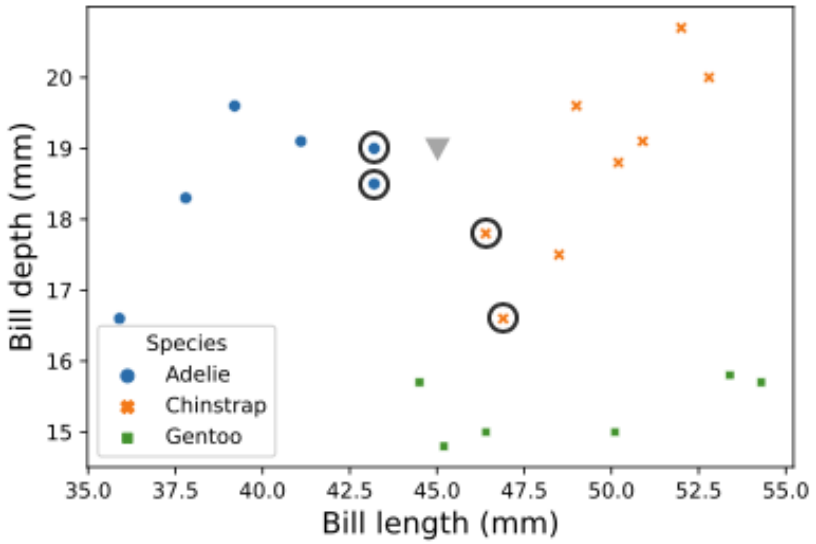
A **hyperparameter** is a user-defined setting in a machine learning model that is not estimated during model fitting.

- Changing the values of a hyperparameter affects the model's performance and predictions.
- k-nearest neighbors is sensitive to two hyperparameters: the value of k and the distance measure. Models with different values of k may result in different predictions.
- Setting k too small results in predictions that are based on only a few instances and thus highly variable.
- But setting k too large often leads to models that are underfit. In practice, k is usually set between 3 and 15 .



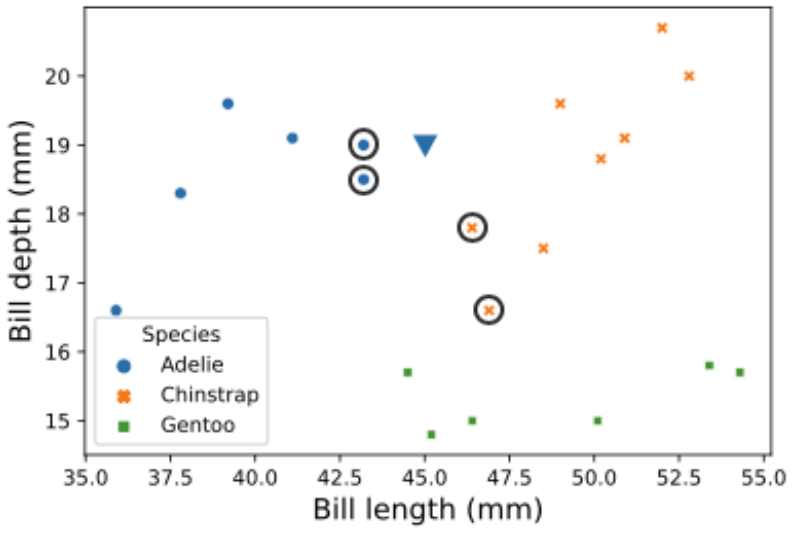
Instance	Distance	Class
6	1.8	Adelie
12	1.9	Adelie
3	1.9	Chinstrap
8	3.1	
16	3.3	
10	3.8	
1	3.9	
14	4.0	
20	4.2	
17	4.2	
7	5.2	
5	5.8	
9	5.9	
19	6.5	
13	7.2	
2	7.2	
11	7.9	
18	9.0	
4	9.4	
15	9.9	

k-nearest neighbors with k=3 classified a penguin at $x^* = (45, 19)$ as an Adelie penguin. But classifications may change depending on the value of k.



Instance	Distance	Class
6	1.8	Adelie
12	1.9	Adelie
3	1.9	Chinstrap
8	3.1	Chinstrap
16	3.3	
10	3.8	
1	3.9	
14	4.0	
20	4.2	
17	4.2	
7	5.2	
5	5.8	
9	5.9	
19	6.5	
13	7.2	
2	7.2	
11	7.9	
18	9.0	
4	9.4	
15	9.9	

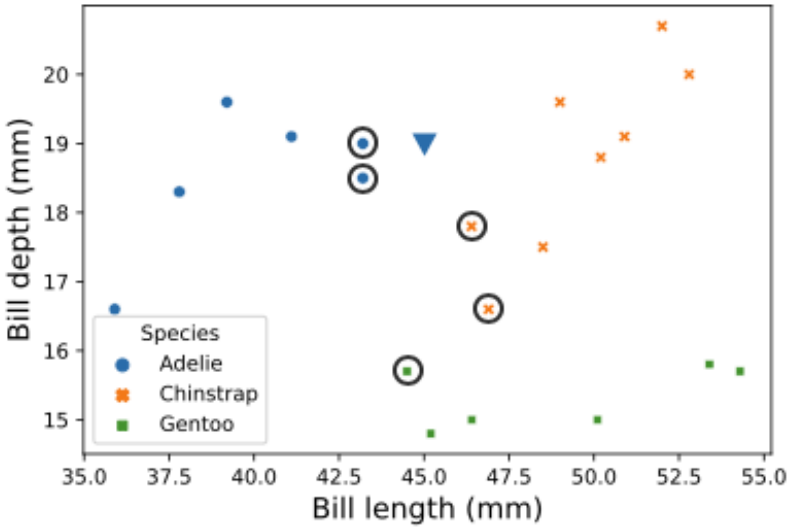
Suppose k=4. Two neighbors are Adelie, and two neighbors are Chinstrap. The classes are tied, so no class is most frequent.



Instance	Distance	Class
6	1.8	Adelie
12	1.9	Adelie
3	1.9	Chinstrap
8	3.1	Chinstrap
16	3.3	
10	3.8	
1	3.9	
14	4.0	
20	4.2	
17	4.2	
7	5.2	
5	5.8	
9	5.9	
19	6.5	
13	7.2	
2	7.2	
11	7.9	
18	9.0	
4	9.4	
15	9.9	

← Nearest neighbor

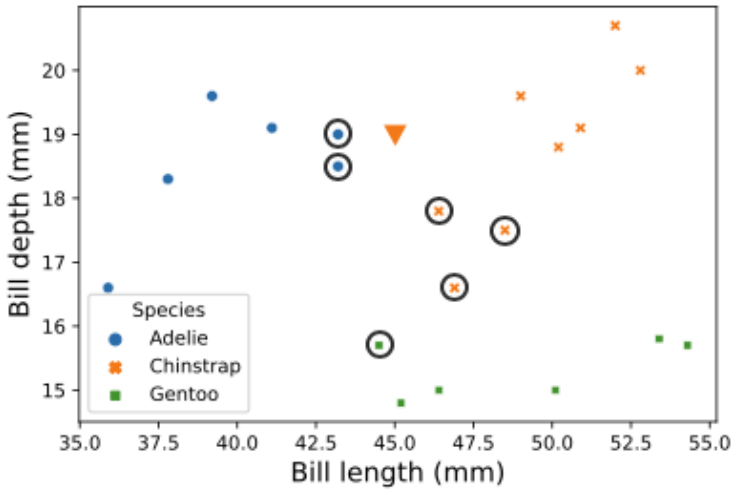
One way to break the tie is by classifying based on the nearest neighbor. Since the nearest neighbor is Adelie, the classification is Adelie.



Instance	Distance	Class
6	1.8	Adelie
12	1.9	Adelie
3	1.9	Chinstrap
8	3.1	Chinstrap
16	3.3	Gentoo
10	3.8	
1	3.9	
14	4.0	
20	4.2	
17	4.2	
7	5.2	
5	5.8	
9	5.9	
19	6.5	
13	7.2	
2	7.2	
11	7.9	
18	9.0	
4	9.4	
15	9.9	

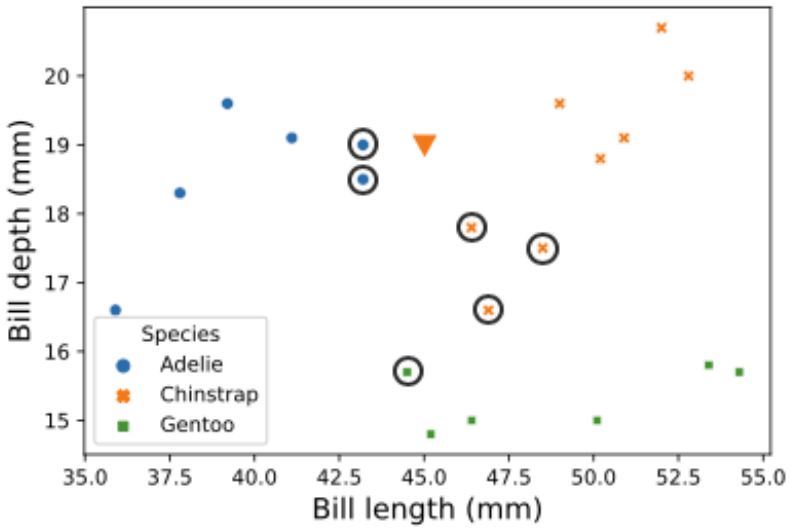
Nearest neighbor

For k=5, two neighbors are Adelie, two neighbors are Chinstrap, and one is Gentoo. A tie still exists, so the classification is based on the nearest neighbor.



Instance	Distance	Class	
6	1.8	Adelie	← Nearest neighbor
12	1.9	Adelie	
3	1.9	Chinstrap	
8	3.1	Chinstrap	
16	3.3	Gentoo	
10	3.8	Chinstrap	
1	3.9		
14	4.0		
20	4.2		
17	4.2		
7	5.2		
5	5.8		
9	5.9		
19	6.5		
13	7.2		
2	7.2		
11	7.9		
18	9.0		
4	9.4		
15	9.9		

At k=6, the tie is broken. The most common class from the nearest neighbors is Chinstrap, so the prediction $\hat{y}^*$ is Chinstrap.



Instance	Distance	Class
6	1.8	Adelie
12	1.9	Adelie
3	1.9	Chinstrap
8	3.1	Chinstrap
16	3.3	Gentoo
10	3.8	Chinstrap
1	3.9	
14	4.0	
20	4.2	
17	4.2	
7	5.2	
5	5.8	
9	5.9	
19	6.5	
13	7.2	
2	7.2	
11	7.9	
18	9.0	
4	9.4	
15	9.9	

← Nearest neighbor

Predictions

k	$\hat{y}^*$
3	Adelie
4	Adelie (tie)
5	Adelie (tie)
6	Chinstrap

The number of instances n and the number of classes c should be considered when selecting k.

Practice Questions

The following table contains the distances, bill length, bill depth, and class of the seven closest instances to a penguin with a bill length of 48 mm and a bill depth of 16 mm.

Distance	Bill length (mm)	Bill depth (mm)	Class
1.3	46.9	16.6	Chinstrap
1.6	48.5	17.5	Chinstrap
1.9	46.4	15	Gentoo
2.3	50.1	15	Gentoo
2.4	46.4	17.8	Chinstrap
3.0	45.2	14.8	Gentoo
3.5	44.5	15.7	Gentoo

1) Using k = 4, how should this penguin be classified?

- ○ Chinstrap
- ○ Gentoo
- ○ Tie exists

2) Using k = 5, how should this penguin be classified?

- ○ Chinstrap
- ○ Gentoo
- ○ Tie exists

3) Using k = 7, how should this penguin be classified?

- ○ Chinstrap
- ○ Gentoo
- ○ Tie exists

4) ____ values of k are recommended to avoid ties.

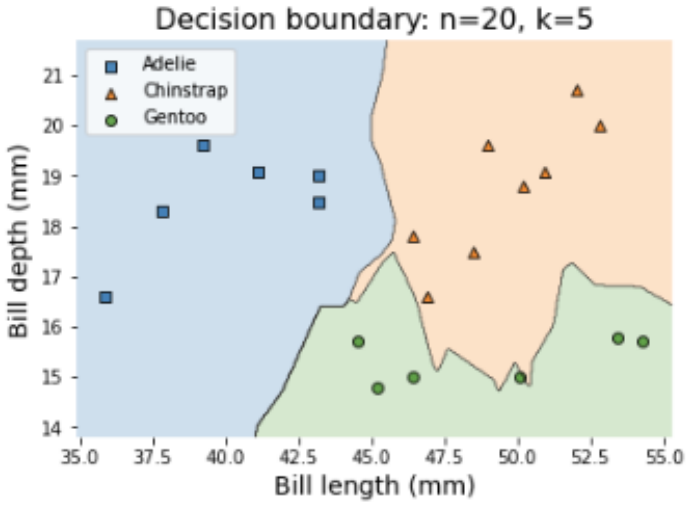
- ○ Even
- ○ Odd

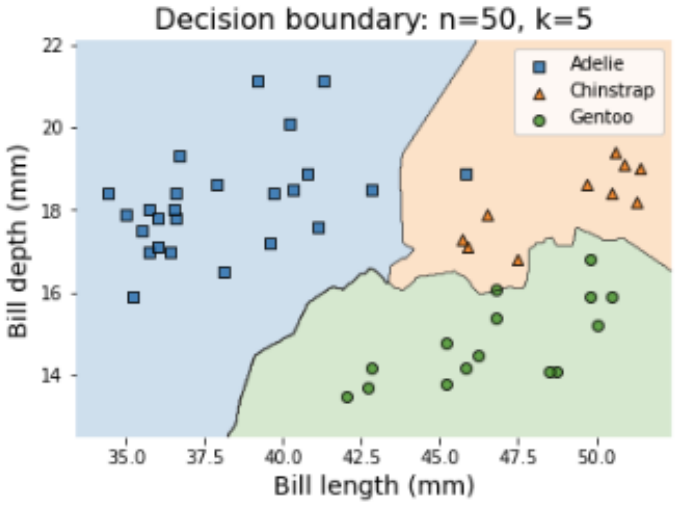
Decision Boundaries

Decision boundaries represent the dividing line between predicting one class vs. another class.

- Decision boundaries may be visualized using a scatter plot for one or two input features, or multiple scatter plots when the number of input features $p > 2$.
- Examining a decision boundary plot helps researchers understand the predictions of a machine learning model and identify a model's strengths and weaknesses.
- Decision boundary plots are also useful tools for comparing models.

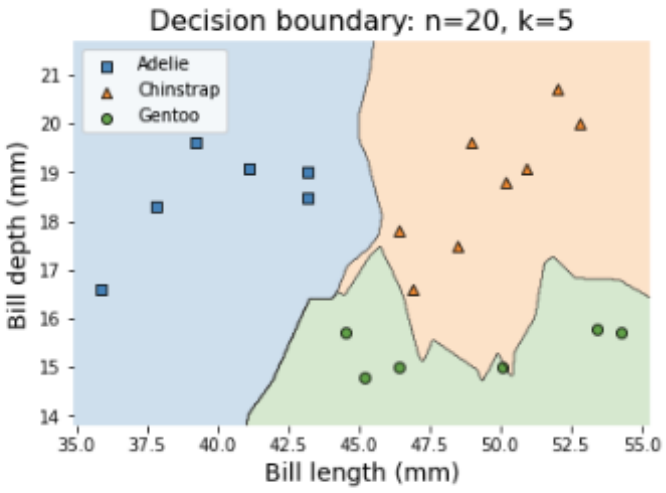
Exploring classification models with decision boundary plots.

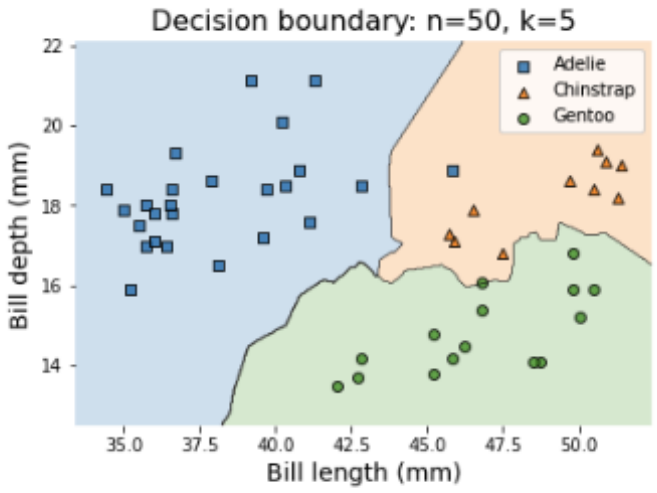


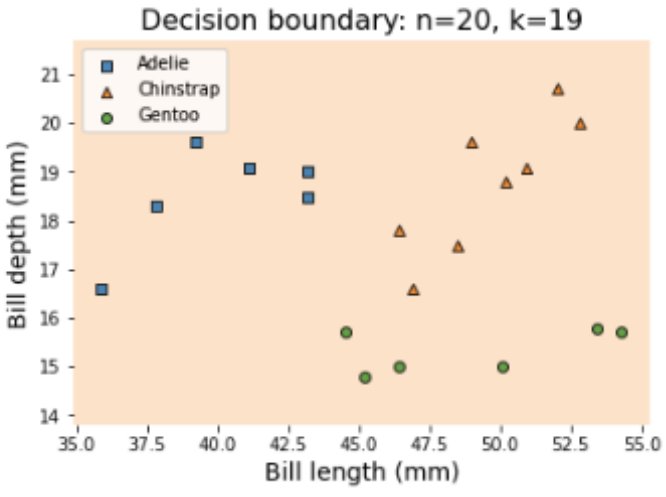


Consider two samples of penguins with n = 20 and n =50. Using k =5, the decision boundary plots both contain three distinct regions.

Exploring classification models with decision boundary plots.

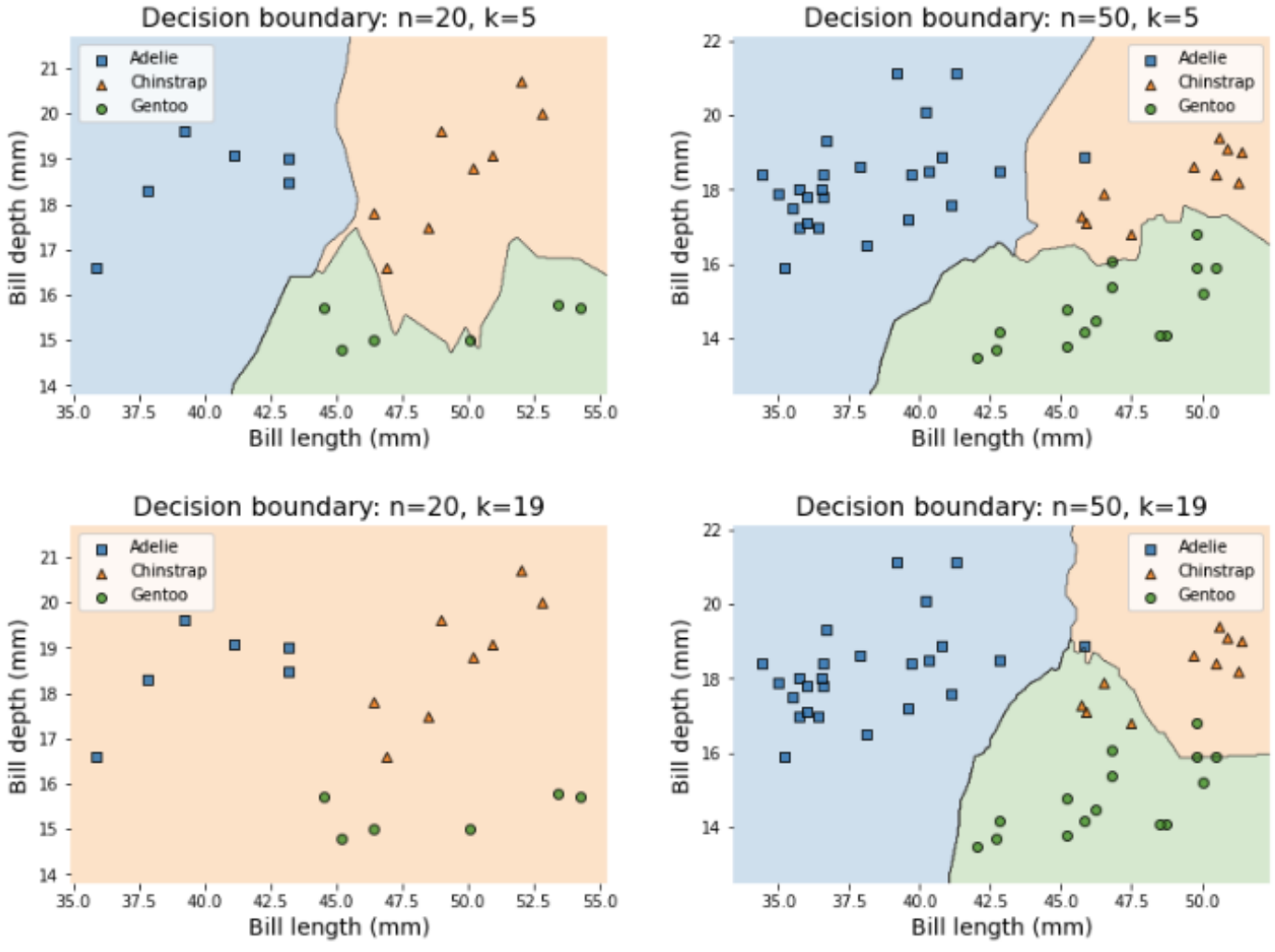






If k is too large, the most common class dominates. For n = 20, the most common class is Chinstrap, so the k-nearest neighbors model with k = 19 only predicts Chinstrap.

Exploring classification models with decision boundary plots.



The larger sample is less sensitive to a larger k.

Practice Questions: Effect of k on decision boundaries.

Use the decision boundary plot animation to answer the following questions.

1) Which value of k results in three distinct, equally sized decision regions for the sample of n = 20 penguins?
   - ○ 5
   - ○ 19

2) Why does the decision boundary plot based on a sample of n = 20 penguins only have one region for k = 19?
   - ○ Almost all instances are included in the nearest neighbors when k = 19.
   - ○ All instances are included in the nearest neighbors when k = 19.
   - ○ All nearest neighbors are Chinstrap for k = 19.

3) How many distinct regions exist in the decision boundary plot based on n = 50 penguins with k = 19?
   - ○ One
   - ○ Two
   - ○ Three

4) How are the number of neighbors k and the sample size n related?
   - ○ No relationship exists.
   - ○ The maximum suggested k depends on n.
   - ○ The minimum suggested k depends on n.

Distance Measures

k-nearest neighbors classifies instances based on the classes of the k closest instances.

- But, depending on how the distance between instances is defined, the nearest neighbors may change.
- Three common distance measures for k-nearest neighbors classification are Euclidean distance, Manhattan distance, and Minkowski distance.

Consider two instances, j and k, with p input features.

- The **Euclidean distance** between instances j and k is

$$d_E(j,k) = \sqrt{\left(x_{1j} - x_{1k}\right)^2 + \left(x_{2j} - x_{2k}\right)^2 + \ldots \left(x_{pj} - x_{pk}\right)^2} = \left(\sum_{i=1}^{p} \left(x_{ij} - x_{ik}\right)^2\right)^{1/2}$$

- **Manhattan distance**, or city block distance, is based on the absolute difference of the coordinates.

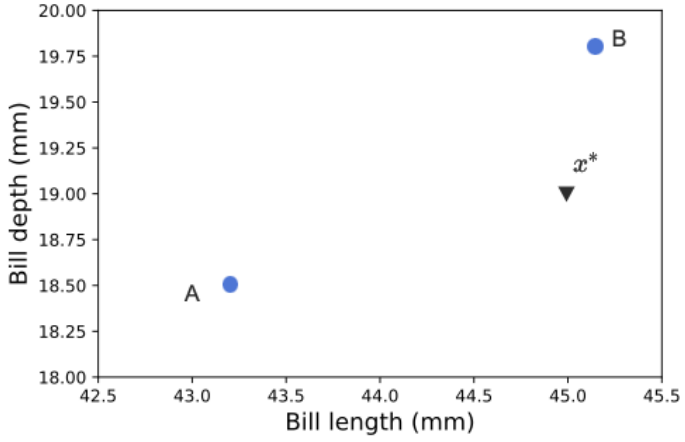
$$d_M(j,k) = \sum_{i=1}^{p} \left|x_{ij} - x_{ik}\right|$$

- **Minkowski distance** is a generalized distance measure, with the power allowed to vary.

$$d_m(j,k) = \left(\sum_{i=1}^{p} \left|x_{ij} - x_{ik}\right|^{1/m}\right)^{1/m}$$

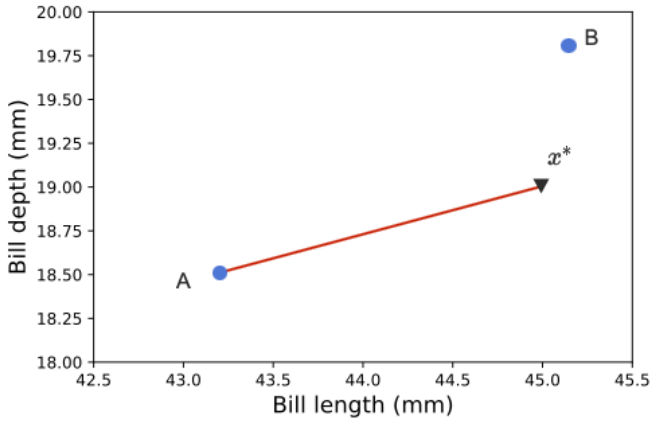
Euclidean distance is equivalent to Minkowski distance with $m = 2$, and Manhattan distance is equivalent to Minkowski distance with $m = 1$. Different situations may call for different distance measures.

Calculating distance measures.



Consider two instances from the penguins dataset: A = (43.2, 18.5) and B = (43.2, 19.1). How close is each instance to $x^* = (45, 19)$?

Calculating distance measures.



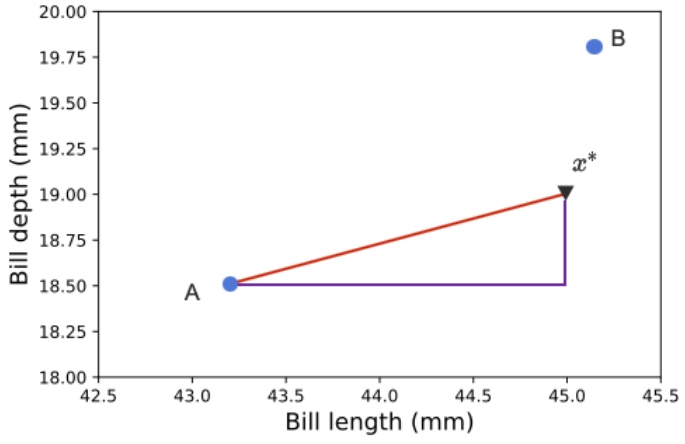
Instance A = (43.2, 18.5)

Euclidean distance $d_E(x^*, A) = \sqrt{(43.2 - 45)^2 + (18.5 - 19)^2} = 1.9$

Euclidean distance represents the straight-line distance between two points. For instance A and x^*,

$$d(x^*, A) = \sqrt{(43.2 - 45)^2 + (18.5 - 19)^2)} = 1.9.$$

Calculating distance measures.



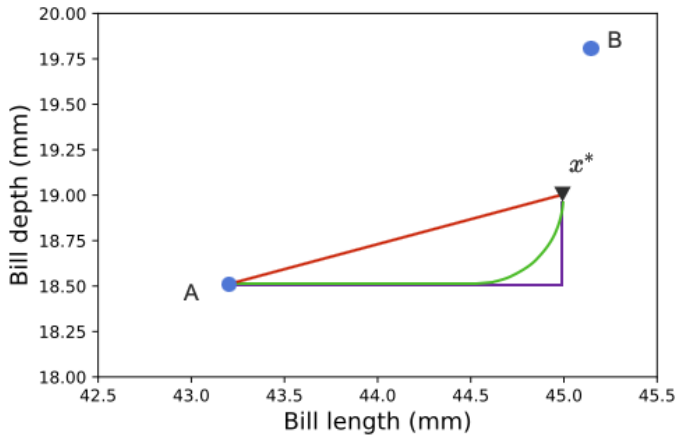
Instance A = (43.2, 18.5)

Euclidean distance $d_E(x^*, A) = \sqrt{(43.2 - 45)^2 + (18.5 - 19)^2} = 1.9$

Manhattan distance $d_M(x^*, A) = |43.2 - 45| + |18.5 - 19| = 2.3$

Manhattan distance represents the absolute distance in each direction. For instance A,
$d(x^*, A) = |43.2 - 45| + |18.5 - 19| = 2.3$.

Calculating distance measures.



Instance A = (43.2, 18.5)

Euclidean distance $d_E(x^*, A) = \sqrt{(43.2 - 45)^2 + (18.5 - 19)^2} = 1.9$

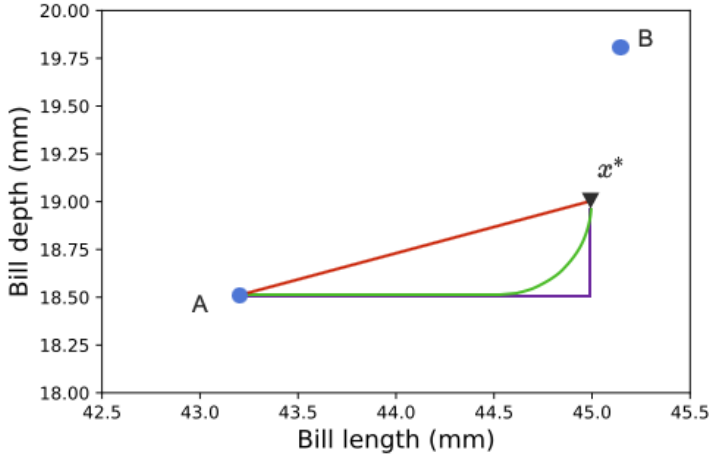
Manhattan distance $d_M(x^*, A) = |43.2 - 45| + |18.5 - 19| = 2.3$

Minkowski distance ($m = 0.5$) $d_m(x^*, A) = \left(|43.2 - 45|^{0.5} + |18.5 - 19|^{0.5}\right)^2 = 4.2$

Minkowski distance can be convex or concave depending on the choice of m. For $m = 0.5$,

$$d(x^*, A) = \left(|43.2 - 45|^{0.5} + |18.5 - 19|^{0.5}\right)^2 = 4.2.$$

Calculating distance measures.



	Instance A = (43.2, 18.5)	Instance B = (45.2, 19.8)
Euclidean distance	$d_E(x^*, A) = \sqrt{(43.2-45)^2 + (18.5-19)^2} = 1.9$	$d_E(x^*, B) = 0.82$
Manhattan distance	$d_M(x^*, A) = \lvert 43.2-45 \rvert + \lvert 18.5-19 \rvert = 2.3$	$d_M(x^*, B) = 1.0$
Minkowski distance ($m = 0.5$)	$d_m(x^*, A) = \left(\lvert 43.2-45 \rvert^{0.5} + \lvert 18.5-19 \rvert^{0.5} \right)^2 = 4.2$	$d_m(x^*, B) = 1.8$

Instance B is closer than Instance A by all three distance measures.

Advantages and Disadvantages

k-nearest neighbors is a flexible classification model that can predict to any number of classes, but limitations exist.

- **Distance-based algorithms** make predictions based only on the most similar instances, and do not consider relationships between input and output features.

Since k-nearest neighbors only uses the input features to identify the nearest instances, k-nearest neighbors should not be used to describe relationships between input and output features.

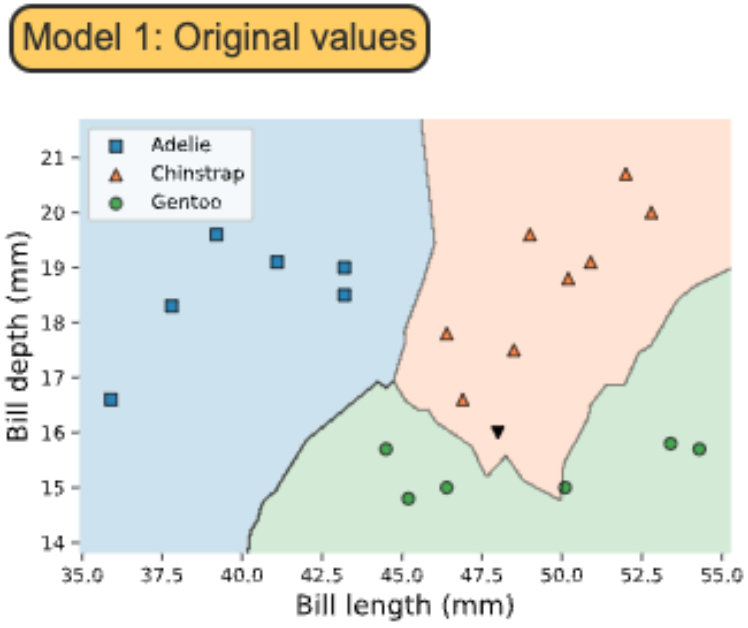
Distance-based algorithms like k-nearest neighbors are sensitive to the unit and magnitude of measurement for each feature.

- Ex: The distance value between the body mass of two penguins depends on whether body mass is measured in grams or kilograms.
- Input features in distance-based algorithms should be standardized before fitting a model. **Standardized features** are scaled to have a mean of 0 and a standard deviation of 1. A feature is standardized by subtracting the mean, $\bar{x}$, and dividing by the standard deviation, s.

  $$z = \frac{x - \bar{x}}{s}$$

  Standardized values are also referred to as z-scores.

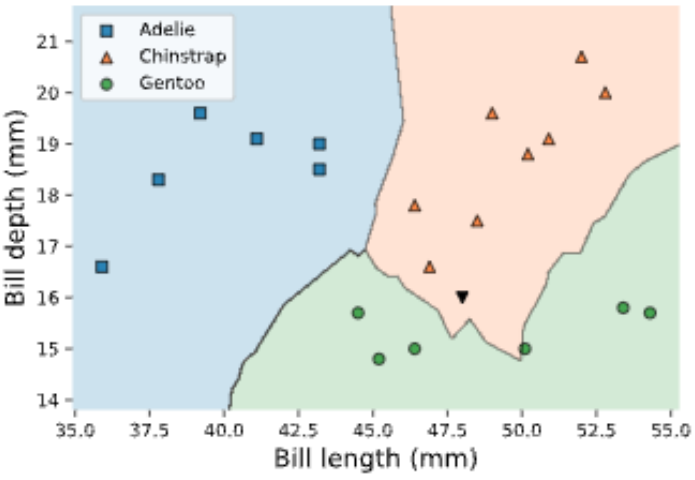
Standardized vs. unstandardized inputs in k-nearest neighbors.



Models based on distance measures, such as k-nearest neighbors, are affected by the input features' scales. Model 1 uses the original input features' values with k=3.

Standardized vs. unstandardized inputs in k-nearest neighbors.

Model 1: Original values

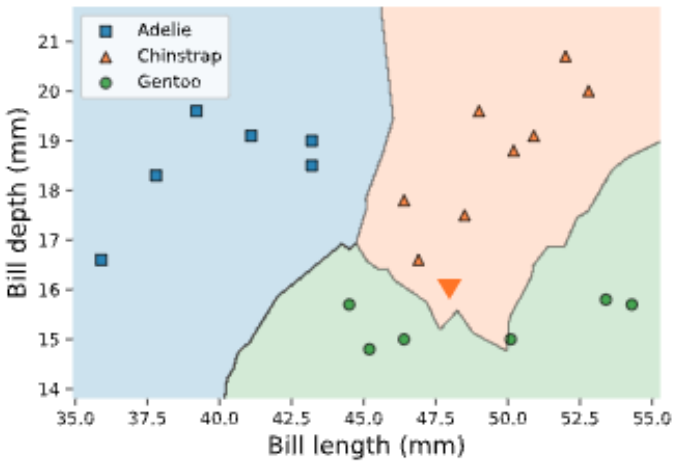


Feature	Mean	SD
Bill length	46.55	5.30
Bill depth	17.66	1.87

Bill length and bill depth are both measured in millimeters (mm). But the mean bill length of penguins is about three times the mean bill depth.

Standardized vs. unstandardized inputs in k-nearest neighbors.

Model 1: Original values

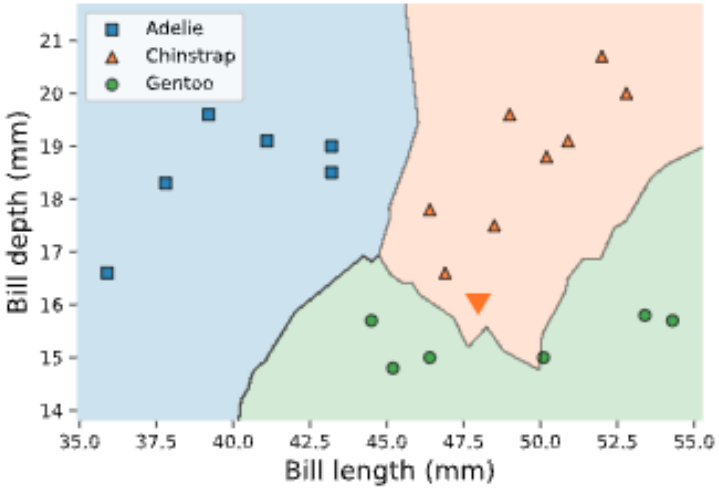


Feature	Mean	SD
Bill length	46.55	5.30
Bill depth	17.66	1.87

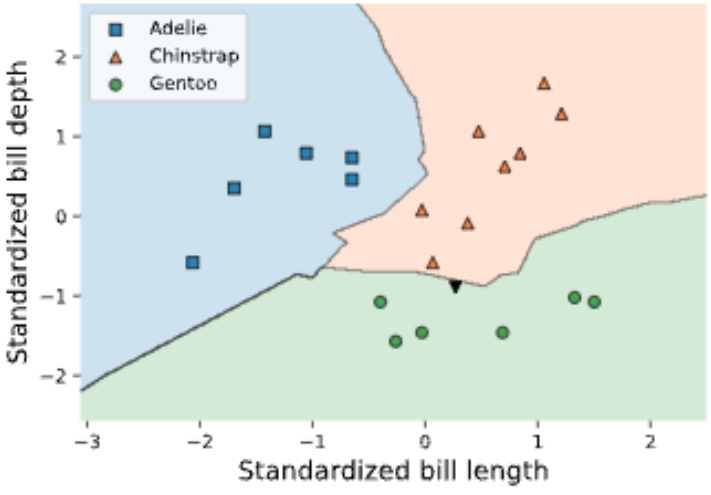
Model 1 classifies a penguin with a bill length of 48 mm and a bill depth of 16 mm as a Chinstrap penguin.

Standardized vs. unstandardized inputs in k-nearest neighbors.

Model 1: Original values



Model 2: Standardized input features

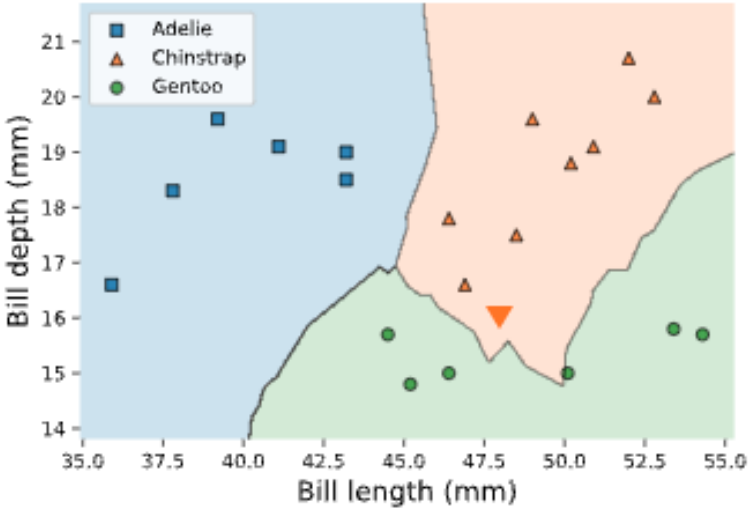


Feature	Mean	SD
Bill length	46.55	5.30
Bill depth	17.66	1.87

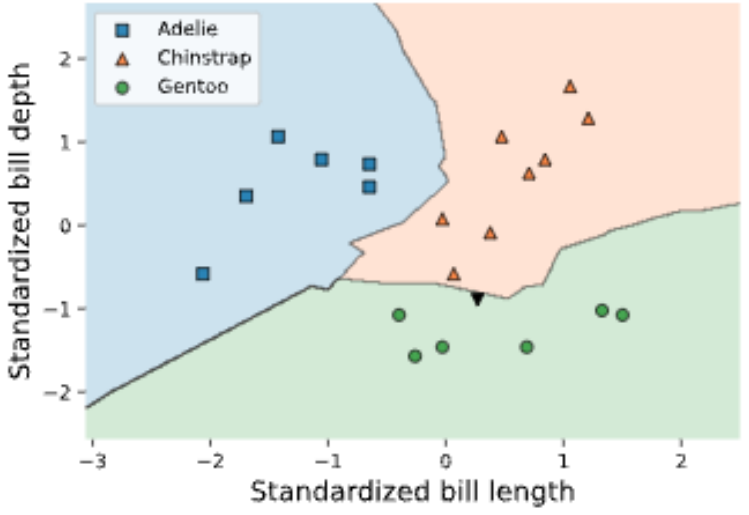
Model 2 also uses k=3, but both input features are standardized first. In standardization, each feature is scaled to have mean 0 and standard deviation 1.

Standardized vs. unstandardized inputs in k-nearest neighbors.

Model 1: Original values



Model 2: Standardized input features



Feature	Mean	SD
Bill length	46.55	5.30
Bill depth	17.66	1.87

Original bill length = 48

Standardized bill length = $\frac{48-46.55}{5.30} = 0.27$

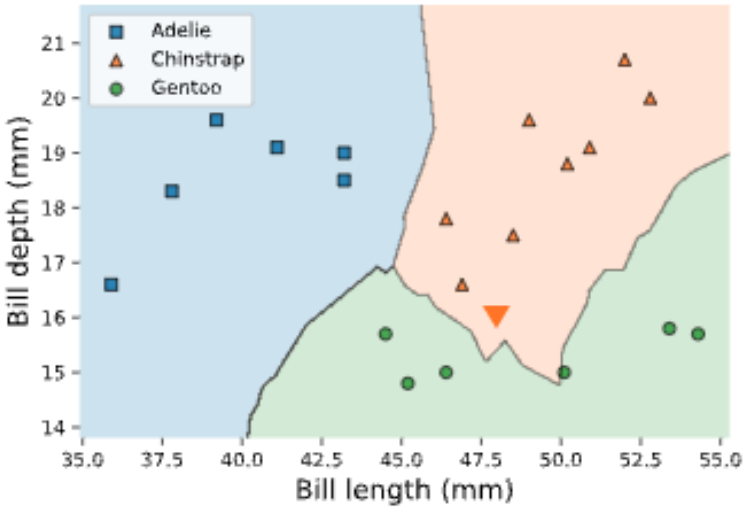
Original bill depth = 16

Standardized bill depth = $\frac{16-17.66}{1.87} = -0.89$

A penguin with a bill length of 48 mm has a standardized bill length of 0.27. A bill depth of 16 mm corresponds to a standardized bill length of -0.89.

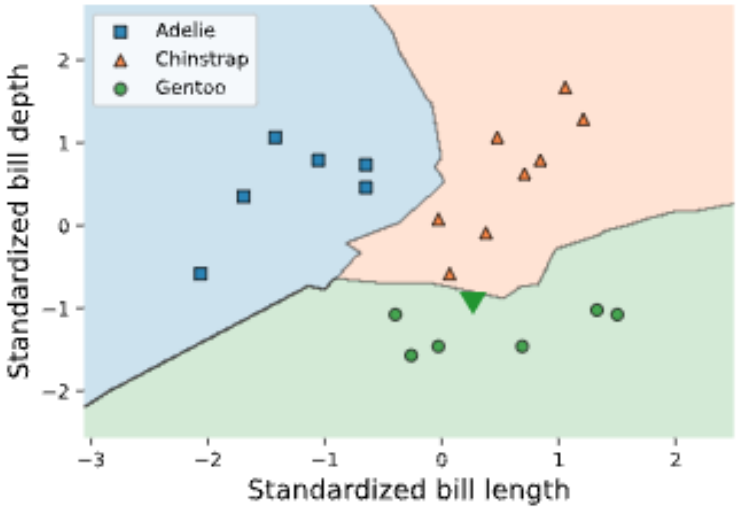
Standardized vs. unstandardized inputs in k-nearest neighbors.

Model 1: Original values



Feature	Mean	SD
Bill length	46.55	5.30
Bill depth	17.66	1.87

Model 2: Standardized input features



Original bill length = 48

Standardized bill length = $\frac{48-46.55}{5.30} = 0.27$

Original bill depth = 16

Standardized bill depth = $\frac{16-17.66}{1.87} = -0.89$

Model 2 classifies the penguin as Gentoo. Using standardized input features results in a different prediction for the same penguin.